# Bayesian Thinking and Linear Classification

APCOMP209a: Introduction to Data Science
Nick Hoernle
Wed 3-4pm & Wed 5:30-6:30 & Thurs 2:30-3:30
nhoernle@g.harvard.edu

## 1 Bayesian Thinking

During the 'hypothesis testing' section of Linear Regression, we saw a philosophy of reasoning where we were presented with a sample from a 'true' underlying statistical process. This sample of data represented a subset of data from the truth and we used hypothesis testing to evaluate the probability of the true parameters being different from a specific null hypothesis given the noisy and incomplete data. The specific observed data was the result of a random sampling procedure from a set of complete data (that would describe the model in its entirety). Bayesian Statistics takes a contrasting approach that dictates there is a distribution over the values that the parameters can hold given the fixed sample of data that we have observed. In this case, we consider the sample of data as fixed and rather try to understand what the specific sample implies for the probability of having a certain model.

We are interested in making conclusions about a model and its associated parameters ($\theta$) given some sample of data ($y$). Under a Bayesian framework, we consider the problem of understanding the probability distribution of the model parameters given the observed data, i.e. $p(\theta|y)$, and to do this, we must first consider the joint probability distribution for $\theta$ and $y$. The joint probability mass or density function can be factored into a multiplication of the *prior distribution* $p(\theta)$ and the *sampling distribution* $p(y|\theta)$. Using Bayes' rule, we can express the *posterior* density in terms of the *prior*, the *sampling distribution* and the *evidence or marginal likelihood* [1] [2]:

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{p(y)} = \frac{p(y|\theta)p(\theta)}{\int_\theta p(y|\theta)p(\theta)d\theta} \quad [1]$$

When the sampling distribution is considered a function of $\theta$ for a fixed $y$ (which in the Bayesian framework is the case) it called the *likelihood function*.

You have seen that when making classification decisions, we can evaluate the *posterior odds* for different parameters $\theta_1$ and $\theta_2$ under a given model as:

$$\frac{p(\theta_1|y)}{p(\theta_2|y)} = \frac{p(y|\theta_1)p(\theta_1)}{p(y|\theta_2)p(\theta_2)} = \frac{p(y|\theta_1)}{p(y|\theta_2)} \times \frac{p(\theta_1)}{p(\theta_2)}$$

which is the same as saying the *posterior odds* are equal to the *prior odds* multiplied by the *likelihood ratio*. This may look familiar from the Bayes' classifier from Logistic regression and we will return to it below when dealing with the LDA and QDA classifiers.

## 2 Bayesian Interpretation of Lasso and Ridge Regression

Refer to Week 3 Advanced Section Notes - Section 5 Bayesian Interpretation of Ridge and Lasso Regression.

[1] here we assume $p(y)$ is continuous and you should use a sum for the discrete case.

# 3 Fisher's Linear Discriminant

In Section 4 we discussed the dimensionality reduction method of Principal Component Analysis (PCA). Similarly, Linear Discriminant Analysis (LDA) can be thought of as a dimensionality reduction technique where a linear discriminant is found that attempts to maximally separate two different classes. PCA, under its assumptions, attempts to find the Principal Components that account for most of the variance in the dataset. On the other hand, LDA attempts to model the difference between the classes of data.[2]

Let's imagine an example in two dimensions with data belonging to one of two classes, where the two classes have (Gaussian) marginal distributions that are highly elongated but aligned (see figure 1 for an example). As you learnt, the first principal component will extract the dimension that captures the highest variance

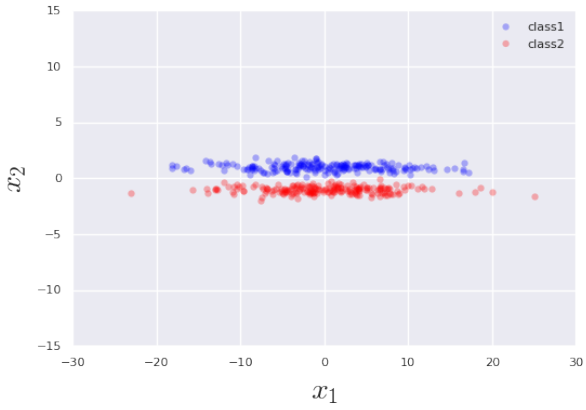


Figure 1: Example dataset where LDA will present a more useful dimensionality reduction than PCA

in the data (in this case it will be exactly $x_1$). For the purposes of dimensionality reduction, projecting the data onto this component will actually result in a one dimensional representation where the data is entirely inseparable (see figure 2).[3]

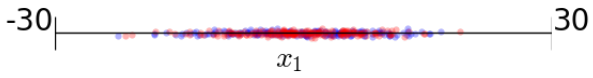


Figure 2: Example of a projection that does not discriminate between the data classes

We can clearly see that for the task of classification, a much more useful projection would be one onto the $x_2$ component. Fisher Linear Discriminant Analysis (LDA) considers maximising an objective with the goal of finding a *discriminating* hyperplane between these classes [3] (and not the hyperplane that describes the variance of the dataset in its entirety). LDA uses the additional information of known class labels to find a more useful discriminator between the data classes. To achieve this, Fisher LDA attempts to project the class

[2]It is worth noting LDA is a supervised technique whereas PCA is unsupervised even though, in this case, we are comparing them for the same function of dimensionality reduction.

[3]For the purposes of visualising the data, I have added a small amount of vertical jitter for plotting the points.

means onto some subspace $\mathbf{W}$ such that the distance between the means is maximised while the variance of the data within each of the classes is minimised. $\mathbf{W}$ can be thought of as consisting of column vectors $\mathbf{w}$ that each represent a hyperplane that discriminates between two classes. To do this, we need the *between-class scatter* matrix [4] $S_B = \sum_k (\mu_\mathbf{k} - \bar{\mathbf{x}})(\mu_\mathbf{k} - \bar{\mathbf{x}})^T$ (measuring the spread of the class means) and the *total within-class scatter* matrix $S_W = \sum_k \sum_{i \in k} (\mathbf{x_i} - \mu_\mathbf{k})(\mathbf{x_i} - \mu_\mathbf{k})^T$. The optimisation task is then to find the vector $\mathbf{w}$ that maximises the following objective (also called the Rayleigh quotient):

$$J(\mathbf{w}) = \frac{\mathbf{w}^T S_B \mathbf{w}}{\mathbf{w}^T S_W \mathbf{w}}$$

Studying the two class case is helpful: let's assume we have data from two Gaussian distributions $(X|Y=1) \sim N(\mu_1, \Sigma_1)$ and $(X|Y=2) \sim N(\mu_2, \Sigma_2)$ for classes 1 and 2 respectively. We have $S_B = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T$ and $S_W = (\Sigma_1 + \Sigma_2)$.

Maximising the objective $J$ with respect to $\mathbf{w}$ is equivalent to maximising the numerator while holding the denominator constant (and as we are only interested in the direction of $\mathbf{w}$, we can hold the denominator constant to 1). So we find $\mathbf{w}$ to $max(\mathbf{w}^T S_B \mathbf{w})$ such that $\mathbf{w}^T S_W \mathbf{w} = 1$ which results in the following Legrangian:

$$L = \mathbf{w}^T S_B \mathbf{w} + \lambda(\mathbf{w}^T S_W \mathbf{w} - 1)$$

Setting $\frac{\partial L}{\partial \mathbf{w}}$ to 0 yields $2(S_B - \lambda S_W)\mathbf{w} = 0$ and so:

$$S_W^{-1} S_B \mathbf{w} = \lambda \mathbf{w} \tag{1}$$

In general the solution exists if $S_W^{-1}$ exists. Moreover, the solution is not unique and corresponds to the eigenvalue problem for the $S_W^{-1} S_B$ matrix.

Referring back to the two class case, and using the definition of $S_B$, we see that

$$S_W^{-1} S_B \mathbf{w} = S_W^{-1}(\mu_1 - \mu_2)(\mu_1 - \mu_2)^T \mathbf{w} = \lambda \mathbf{w}$$

and finally noting that $(\mu_1 - \mu_2)^T \mathbf{w}$ is a scalar we get:

$$S_W^{-1}(\mu_1 - \mu_2) = \frac{\lambda}{\alpha} \mathbf{w} \tag{2}$$

Again, we are only interested in the direction of $\mathbf{w}$ (and can discard the scalar multiplier), so

$$\mathbf{w}^* = S_W^{-1}(\mu_1 - \mu_2)$$

In the more general case where we have $K$ classes (and $K$ associated means $\mu_k$), with $K \geq 2$, this space can only span $K - 1$ dimensions. In otherwords, we can, in theory, project any datapoint $x$ onto the $K - 1$

[4] there are $k$ classes and $i \in k$ is the index of a datapoint belonging to the the $k^{th}$ class. Moreover, $\mu_k$ is the mean of class $k$, $x_i$ is a datapoint and $\bar{x}$ is the mean of the entire dataset.

dimensional space spanned by the $K$ centroids (i.e. classify the data to one of the classes) and not lose any information. Moreover, studying $rank(S_W^{-1}S_B) \leq \min\{rank(S_W^{-1}), rank(S_B)\}$, [4] and asserting that $rank(S_B) < K-1$, we have $rank(S_W^{-1}S_B) \leq K-1$. So LDA projects a p dimensional vector onto a $K-1$ dimensional subspace for which the data is maximally (linearly) separated.

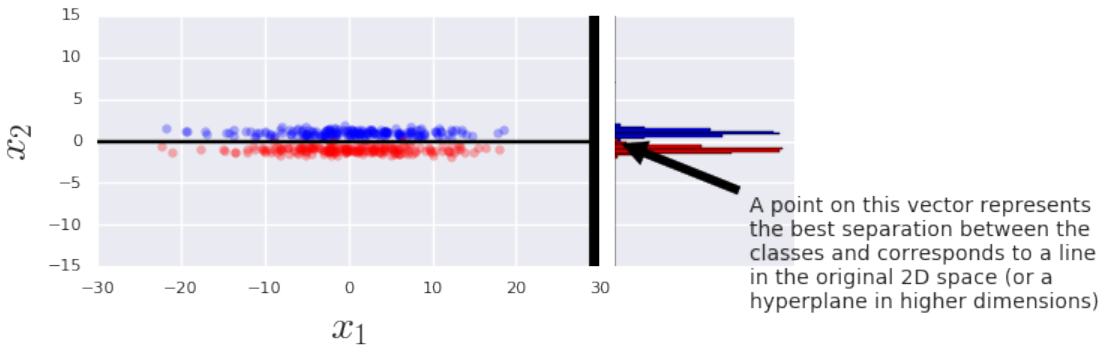


Figure 3: Example of a projection that does discriminate between the data classes

# 4 LDA and QDA Classifiers

We can now tackle the same problem of discriminating among the data classes, but use properties of Gaussian distributions and the posterior odds of a datapoint originating from a certain class specific distribution. From classification Decision theory, we know that to minimise the misclassification rate, it is optimal to classify a datapoint to the most probable class using the posterior probability $P(Y|X)$, where $Y$ denotes the classification decision and $X$ denotes the predictors in the model (refer to the *Bayes Classifier* discussed in class). If we let $f_k(x)$ denote the class-conditional density of X in the class Y=k, and we let $\pi_k$ be the prior probability that a datapoint chosen at random will be observed as class k (note that $\sum_{k=1}^{K} \pi_k = 1$) then we have:

$$p(Y = k|X = x) = \frac{p(X = x|Y = k)p(Y = k)}{p(X = x)} = \frac{f_k(x)\pi_k}{\sum_{l=1}^{K} f_l(x)\pi_l}$$

the denominator follows from the law of total probability. We make the explicit assumption that the densities $f_k(x)$ follow multivariate Gaussian distributions (i.e. $(X|Y = k) \sim N(\mu_k, \Sigma_k)$) which means that the densities follow:

$$f_k(x) = \frac{1}{(2\pi)^{p/2}|\Sigma_k|^{1/2}} \exp\left\{\frac{-1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k)\right\}$$

Classifying a new datapoint to the class with the highest posterior probability, $max(p(Y|X)) = max(\log[p(Y|X)])$, results in classifying the datapoint to the class that maximises:

$$\delta_k(x) = log(\pi_k) - \frac{1}{2}log(|\Sigma_k|) - \frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k)$$

Notice that $(x-\mu_k)^T\Sigma^{-1}(x-\mu_k)$ is the squared Mahalanobis distance metric. Further note that for the two class problem (with class 1 and class 2), $\delta_1(x)=\delta_2(x)$ represents the decision boundary between the two classes, and this decision boundary is quadratic in $x$ (*refer to your lecture notes for more on this topic*).

Dealing with the case that the classes have the same (pooled) covariance matrices, we have the objective of classifying the datapoint to the class that maximises:

$$\delta_k(x) = x^T\Sigma^{-1}\mu_k + log(\pi_k) - \frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k$$

Again, for the two class problem, we can look at the log posterior-odds:

$$\frac{p(Y=k|X=x)}{p(Y=l|X=x)} = \frac{f_k(x)\pi_k}{f_l(x)\pi_l} = log\frac{\pi_k}{\pi_l} - \frac{1}{2}(\mu_k+\mu_l)^T\Sigma^{-1}(\mu_k-\mu_l) + x^T\Sigma^{-1}(\mu_k-\mu_l)$$

Note now that the decision boundary would be linear in $x$. Further note that the $\Sigma^{-1}(\mu_k-\mu_l)$ direction term is the same as the $\mathbf{w}$ vector found above.

# 5 Notes

1. LDA/QDA classifies a new datapoint to the class with the closest centroid. Here 'closest' is measured in the Mahalanobis metric, using a pooled covariance estimate.
2. Under the assumption that the data are multi-variate Gaussian within each class, LDA is a Bayes' classifier (i.e. it minimises the probability of making a mis-classification).
3. LDA results in linear decision boundaries.
4. In many cases, when linear decision boundaries are inadequate for separating the classes, QDA can be used, with the cost of needing to estimating more parameters.
5. LDA needs to approximate the following $(K+K+P^2)$ statistics from the data[5]:
   (a) K priors: $\hat{\pi}_k = \frac{N_k}{N}$
   (b) K means: $\hat{\mu}_k = \sum_{i\in k}\frac{x_i}{N_k}$
   (c) 1 covariance matrix: $\hat{\Sigma} = \sum_{k=1}^{K}\sum_{i\in k}\frac{(x_i-\hat{\mu}_k)(x_i-\hat{\mu}_k)^T}{N-K}$
6. QDA needs to approximate the following $(K+K+K\times P^2)$ statistics from the data (notice the possibility for overfitting your training data here):
   (a) K priors: $\hat{\pi}_k = \frac{N_k}{N}$
   (b) K means: $\hat{\mu}_k = \sum_{i\in k}\frac{x_i}{N_k}$
   (c) K covariance matrices: $\hat{\Sigma}_k = \sum_{i\in k}\frac{(x_i-\hat{\mu}_k)(x_i-\hat{\mu}_k)^T}{N_k}$

Other interesting and worthwhile references are: [5], [6], [7].

[5] Here I use $i\in k$ to denote the index of datapoint $i$ that belongs to class $k$.

# References


[1] A. Gelman and C. R. Shalizi, "Philosophy and the practice of bayesian statistics," *British Journal of Mathematical and Statistical Psychology*, vol. 66, no. 1, pp. 8–38, 2013.

[2] A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin, *Bayesian data analysis*, vol. 2. CRC press Boca Raton, FL, 2014.

[3] M. Welling, "Fisher linear discriminant analysis," *Department of Computer Science, University of Toronto*, vol. 3, no. 1, 2005.

[4] K. B. Petersen, M. S. Pedersen, *et al.*, "The matrix cookbook," *Technical University of Denmark*, vol. 7, p. 15, 2008.

[5] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on*, pp. 586–591, IEEE, 1991.

[6] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 19, no. 7, pp. 711–720, 1997.

[7] A. M. Martínez and A. C. Kak, "Pca versus lda," *IEEE transactions on pattern analysis and machine intelligence*, vol. 23, no. 2, pp. 228–233, 2001.